First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

□ Generate Collection Print

L14: Entry 1 of 10 File: USPT Sep 14, 2004

DOCUMENT-IDENTIFIER: US 6791582 B2
TITLE: Method and system for identifying and displaying information that is new or has been updated in a place

Application Filing Date (1):
20001229

Brief Summary Text (7):
In order for such disperse teams to have the same, or substantially the same, collaboration environment as individuals working in the same physical office, a system is required which facilitates instant messaging, voice conferencing, electronic white boarding, and text and non-text file exchange. Such a system needs to provide a collaborative electronic room, or space, which is easily configured for use by team members without substantial administrative or application development support, and preferably include both groupware and project oriented applications such as shared folders, file exchange, workflow, group calendars, threaded conversations, version control, file locking, file merging, and security.

Detailed Description Text (5):
QuickPlace open storage 130 includes all the databases and templates that implement the collaboration space. Domino 132 and active directory 134 define the collaboration process. The user interfaces with the system through browser 112. NSF agents 114, 116, Java 118 and LotusScript 120 represent components and templates downloaded from server 100 in support of collaboration space at client 102. All the extensions 108 are keyed off the URL, as will be further explained hereafter.

Detailed Description Text (20):
Referring further to FIG. 3, a preferred implementation of the object model heretofore described uses Lotus Notes/Domino concepts and objects. Thus, Notes/Domino file system directory 202 represents place 172; database 204 represents room 174; folder view 206 represents folder 176; pages 182, members 190, forms 178, fields 180 and skins 200 are represented by notes 208, 210, 212, 214, 220, respectively.

Detailed Description Text (21):
Place 172 is represented as a file system directory. So whenever a place called Acme is created, a file system directory 202 called Acme will be instantiated. Within that directory 202, there are a number of objects. Each room 174 is a Notes database 204. Folders 176 are implemented as Notes folders or views 206 depending on what's more convenient for the particular folder.

Detailed Description Text (24):
Developers familiar with the Domino Object Model (Domino OM) will be able to leverage their existing skills when developing on the QuickPlace platform. "PlaceBots" for example are actually implement Domino Agents, and it is possible to create and test them on Domino Databases. Within the QuickPlace object model (OM), however, there are some divergences from the Domino OM. For example, QuickPlace forms 178 are not the same as Domino Forms. QuickPlace forms more closely resemble Domino Documents, because they are created using a Domino form, and contain a Domino text field with a value of "h_Form". The value of "h_form" tells QuickPlace

that this Domino document should be rendered in a browser as a QuickPlace form 178.

Detailed Description Text (26):
QuickPlace forms 178 have been optimized by stripping away many of the Notes features not required when used on the Web. A another advantage of this structure is that it enables the use of Web authoring tools to extend the objects. For example, with respect to QuickPlace forms, it is possible to modify forms using XML, JavaScript and HTML and any other Web tools. Knowledge of JavaScript and HTML are more common than Domino Designer skills, thus making the QuickPlace a very open platform. Some parts of the QuickPlace OM implement Domino/Notes functionality in different ways to a standard Domino application. For example, QuickPlace uses Domino's security and authentication model as a basis for its management of access to QuickPlaces. However, instead of primarily utilizing the Domino Directory, QuickPlace also uses a Contacts1.nsf database for each QuickPlace.

Detailed Description Text (37):
To locate QuickPlace Server 170 in PlaceBots and get access to all of its databases, a LotusScript method GetDbServer is executed together with a test that the Path to databases starts with "QuickPlace".

Detailed Description Text (39):
Place object 172 is a directory in the "QuickPlace" directory grouping resources for a Place. The Domino equivalent is a file directory bearing the name of the QuickPlace. Place object 172 is a directory that brings together a Place for organizational purposes. It also identifies the NSFs 114 as belonging to the place 172 by bearing the name of the QuickPlace. As distinguished from a place object 172, the main room 174 in a QuickPlace is a database called Main.nsf. Place object 172 groups and identifies the Main.nsf resources for the Place and any subrooms 194 in the Place. Place object 172 contains several files. There is a Main.nsf, Contacts1.nsf and a Search.nsf file. If the QuickPlace has a Subroom 194 there will also be an NSF file with its name starting with "PageLibrary". Each of these page library files is a Room 174.

Detailed Description Text (40):
The place object in PlaceBots 184: place object (directory) 172, contains the databases which form a place. When writing PlaceBots, one can search for this directory by using the name of the QuickPlace. In this directory will be found all the databases will belong to that place 172. This file directory's name is the name of the QuickPlace. For example, if the QuickPlace is called "Millennia", this directory has the following path within the QuickPlace server 170:

Detailed Description Text (44):
Room object 174 is the main container for a Place, containing a collection of pages and tools. The Domino Equivalent an NSF Database. The room 174 is the main container for a QuickPlace's content. For example, when using the Millennia Place, most of what is seen is contained in the Room object. The Room object is always called Main.nsf, and holds folders 176 and pages 182 for the QuickPlace, as well as managing links to any subrooms 194 in the place object 172. Room object 174 uses elements held in other databases. For example many of the standard images QuickPlace displays are in a resources object (not shown). Each room 174 has its own security and authentication, and the information required to do this is contained in databases such as Contacts1.nsf. A room 174 breaks down a place 172 into smaller areas to help define structure. Each room 174 has its own security and authentication. This allows separate user groups. It also means that subrooms 194 can be created for separate projects, forming a separate shared space. The room object 174 then forms a common entry point where shared resources can be stored.

Detailed Description Text (45):
The room object in PlaceBots: to locate a room 174, one looks in the main

QuickPlace Server 170 directory, then looks into the room object (a directory bearing the name of the QuickPlace), then looks for a database called "Main.nsf".

Detailed Description Text (107):
A subroom object 194 is a container within a QuickPlace with separate security to main Room. The Domino equivalent is an NSF Database. Subrooms 194 are similar in structure to Rooms and are used to create discreet meeting places for subset of the Members in a Place.

Detailed Description Text (109):
The Instr method has been used to look for this database, down to the PageLibrary part of the string, because it is difficult to know what the 16 digit number will be.

Detailed Description Text (112):
A resources object (not shown) is database of shared resources, having as its Domino equivalent NSF Database. It serves as a centralized container for resources required in all QuickPlaces on a server. Images, layout files and fonts are stored in this database. For example resources such as the button that appears beside the simple search image "Go.gif" is stored in this database. The easiest way to find items in this database is by scrolling through the h_SystemNameView. A dummy form may be used to view such elements.

Detailed Description Text (119):
An example of locating a file without opening it is when a QuickPlace skin accesses a JavaScript LSS file. The user never sees the LSS page, but its contents are used by the visible page to render objects and perform functions. To locate a document in Domino, the initial part of the URL is pointed to the host server, then the database containing the required document. The next part of the URL must point to a view with the first column specified as being sorted. This first, sorted column becomes the key column. Then a URL is used to open the document, as in the following example:

Detailed Description Text (120):
http://Host/Database/View/Key?DominoURLCommand

Detailed Description Text (127):
The structure of URLs in a QuickPlace is the same as in any Domino database. QuickPlace objects are quite often referred to via relative URLs. For example, to reference a page that has been created, the following syntax is used: ././h_View/PageName?OpenDocument Where: "././" section at the front of the URL creates a relative URL, is interpreted by the Domino server as referring to the parent objects of the current object (h_View and PageName).

Detailed Description Text (141):
In accordance with a further embodiment of the invention, selection of quick browse opens a new window and creates a set a links to control the main window without losing the context. Context appears in a separate window in a simpler HTML format. Selection of an item in that separate window, causes the page or room (if link is to a room) to be displayed in the main window. QuickBrowse is a window that can be popped up by clicking a remote control icon in search results and in whatsnew. It enables users to browse links generated by these functions in a random-access order. When the user clicks a link in the quickbrowse window, the main window is reloaded with the target page. This is distinguished from Quick search, which is the search field and green button that is embedded in QuickPlaces like in most web sites, so that users can enter a search term and perform a search in a single click--i.e., without first having to go to a special search page.

Detailed Description Text (167):
when in stand-alone mode. It contains the following files: Main.nsf, Contacts1.nsf,

CreateHaiku.nsf, Admin.nsf. The templates for these Domino databases reside in the directory named AreaTypes. The "Welcome" page may be configured to suit the needs of an organization.

Detailed Description Text (172):
In accordance with a preferred embodiment of the invention, a user may receive a report of what is new, either scheduled or on request, such as by clicking on a place or an E-mail component. What ever is selected, the displayed result is personalized to the individual user, including security. The scope of the what's new page may be, for example, user specified as daily or weekly.

Detailed Description Text (175):
Referring to FIG. 11, each room in collaboration space is implemented as a database 310-312. A Notes view 314-316 and folder 318-320 is associated with each such database. Using notes security, each person looking at view sees what is authorized to see. Folder 318-320 is updated every night, for example, by taking a snap shot of view 314-316. This update can also be done upon request, which will cause view 314-316 to be immediately copied to folder 318-320. When each user request for what's new, filtering on user authorization and scope occurs.

Detailed Description Text (178):
Referring to FIG. 11, a place comprises a plurality of rooms, each room represented, respectively, by a room database 310-312. Each has a view 314-316, respectively, which views are maintained up to date per formula. These views, when updated, such as every night and whenever someone updates, are copied into respective folders 318-320. When a user asks what's new in step 328, in step 322 QP server 100 opens the contents of folders 318-320, in step 324 processes the data, using access control list and other Notes security, and in step 326 generates HTML from that data to reorganize it for display 330. Step 326 html is created with all links and, in a preferred embodiment of the invention, quick browse feature 304 in window 330 which, upon user selection, generates display 332 presented at browser 112. This is just the on-line part. Off line, step 326 is create and send mail (not html), which it does on behalf of each member of the room. User selection of quick browse 304 opens a new window 332, creates a set a links to control the main window 112 without losing the context. Context appears in a separate window 332 in a simpler html format. Selection of an item in that separate window 332, causes the page or room (if link is to a room) to be displayed in the main window 112.

Detailed Description Text (179):
In accordance with a further embodiment of the invention, a team version of a what's new newsletter can also be collected in the QuickPlace, enabling users to browse a set of changes efficiently, and to review past newsletters. Users can also click quick-search links to answer common queries such as "what's changed in this QuickPlace today?".

Detailed Description Text (189):
Referring to FIG. 14, in accordance with a preferred embodiment of the invention, quick-browse is implemented as a "remote control" UI mechanism 304. A Remote Control link/icon is displayed in any context 296 that supports it, such as Search Results and What's New. When the user clicks the remote control link 302, such as button 536 (FIG. 10), or 532 (FIG. 9), a window 296, 330 opens with a compact set of links 304 to browse. When the user clicks a link in the remote control 304, the main QuickPlace window 330 is redirected to the linked page 332. Other contexts such as the Tutorial and Help may themselves act as specialized versions of the remote control by opening into a separate window. For example, the tutorial is implemented as a set of pure HTML files which can be customized/switched by third parties. Some links from the tutorial may use the same mechanism to open links in main window.

Detailed Description Text (190):

Clicking the remote control icon opens a quickbrowse window, containing one or more links. Upon being clicked, these links open in the main QuickPlace window; therefore the QuickBrowse window acts like a remote control onto the main window. This allows the set of hits to remain available as the user browses around, in any order. What's New and Search Results generate links that can be anywhere in a QuickPlace--i.e., in any of possibly many different rooms. The QuickBrowse window is an attempt to bring some coherence to the experience of jumping around all over the QuickPlace.

Detailed Description Text (197):
To start a chat session with one other QuickPlace member, a member clicks "chat" 390 (FIG. 12) under "Tools" in the sidebar. Referring to FIG. 13, in the "Who is here?" section of the QuickPlace Team Chat window that now appears on the screen, the member double-clicks the name of the person with whom to chat. In the box 538 labeled "Type your text" in the new window that appears on the screen, the member types the text of the message to be sent. Pressing Enter or clicking send sends the message. The message is sent and displayed in the chat transcript area 540. When the member receives a reply to the message, the reply appears in the same area 540. The chat session may be terminated by clicking close or selecting close from a message menu. To remain available for future chat sessions with other QuickPlace members, the QuickPlace Team Chat window 540 is left open or minimized. To dismiss the QuickPlace Team Chat window, the close box for the window is clicked. To initiate another chat session or receive invitations to join chat sessions, it is necessary to reopen the QuickPlace Team Chat window (by clicking "chat").

Detailed Description Text (198):
To start a chat session with two or more QuickPlace members, a member first invites those other members to participate in the session. To create an invitation for two or more members and then start the chat session, the member clicks "chat" 390 in the "Tools" sidebar. In the "Who is here?" section 542 of the QuickPlace Team Chat window that appears on the screen, the member presses and holds theshift or Ctrl key while clicking the names of the members with to chat. Then, a single-option menu may be selected in which to create an invitation for the chat session entering the subject of the chat in the Topic box, entering the text of invitation in the Message text box, optionally selecting "Secure messages" to prevent anyone outside the QuickPlace from intercepting and reading the messages. The member enters and sends messages and receives replies as above.

Detailed Description Text (200):
When a member of a QuickPlace invites a second member to join a chat session, an invitation dialog box appears on your screen of the second member. Using the buttons in the dialog box, the second member can respond privately, join the chat, or decline the invitation. To respond privately--that is, to reply to the sender of the invitation only--click Respond. To accept the invitation, click Join. To decline the invitation and close the invitation dialog box, click Close.

Detailed Description Text (201):
One of two or several participants in a chat session can leave the chat session at any time by clicking Leave or selecting Leave from the Message menu.

Detailed Description Text (202):
Upon signing in, the user can quickly get an overview of who is currently signed in to the QuickPlace. In accordance with an exemplary embodiment of the invention, this is accomplished by displaying a list of names in a separate floating window 542. A user is enabled to control (a) whether or not the chat features actually load on sign-in, so that on low-bandwidth connections in the sametime components don't have to be loaded; and (b) her personal availability state--i.e., whether or not other people can see and chat with her. One UI option is to provide a "Do Not Disturb" button (e.g., in the pathbar)--which logs the user out of the sametime server, turns off all chat features, and toggles itself to display a button labeled

"Who Is Here?". To prevent "lurking", the awareness display may be reciprocal--i.e., if user A can see user B, then user B can see user A. The activation state of the sametime features may be kept persistent across sessions: if the user closes the browser with the chat session turned off, then opens another browser window later, the chat session is reopened in its "off" state. In QuickPlace usage it may be desirable provide awareness of who is in the user's current room, versus the entire QuickPlace. A user is able to leverage the current availability of other members to start a chat or other form of communication with them (e.g., phone call). Chats begin by clicking the user's name or via a menu item in that control. The most important type of chat is N-way private chat by invitation--i.e., the user invites one or more people to join in a private chat which other members cannot see and cannot find out is taking place. Users may be allowed to copy the transcripts of a chat into a QuickPlace page, or to save the chat as a whole as a page. An alternative type of chat is a place chat, which in QuickPlace by default is the whole QuickPlace, and may be made to support persistence--i.e., if may be rendered capable of loading the prior transcript when the user opens the chat window.

Detailed Description Paragraph Table (1):
TABLE 1 QUICKPLACE OBJECTS AND DOMINO EQUIVALENT OuickPlace Object Domino Equivalent QuickPlace Server File Directory Place File Directory Page Data Note, Form & Subform PlaceBot Domino Agent Theme Data Note Member Data Note in Contacts1.nsf Subroom * NSF Database SubroomTheme Data Note SubroomMember Data Note in Contacts1.nsf Room NSF Database Folder/TOC Folder or View Form Data Note Field Data Note * Subrooms contain their own set of Folder, Page, Form, Field, PlaceBot & Subroom Objects

Detailed Description Paragraph Table (2):
TABLE 2 SCRIPT PROCEDURE FOR FINDING A PLACE Dim ndbPlace As NotesDatabase Set dirPlace = New NotesDbDirectory(g_sServerName) Set ndbPlace = dirPlace.GetFirstDatabase(DATABASE) sNdbPlaceFilepath = ndbPlace.FilePath If Instr(1, Lcase(sNdbPlaceFilepath), .vertline. quickplace.backslash.millennia .vertline.) Then '//the Place is found

Detailed Description Paragraph Table (3):
TABLE 3 SCRIPT PROCEDURE TO FIND A ROOM OBJECT Set dirPlace = New NotesDbDirectory (g_sServerName) Set ndbPlace = dirPlace.GetFirstDatabase(DATABASE) sNdbPlaceFilepath = ndbPlace.FilePath If Instr (1, Lcase(sNdbPlaceFilepath), .vertline. quickplace.backslash.millennia.backslash.main.nsf .vertline.) Then '//the Room id found.

Detailed Description Paragraph Table (4):
TABLE 4 FIELDS DEFINING ROOMS Field Name Description h_HaikuName The name of this Place h_AreaType The name of the template used to create this room. h_AreaParent The name of the parent database h_ShowSecurity If h_SetSecurity = 1, the QuickPlace server sets h_ShowSecurity to 1. h_SetCalendar Determines if the Calendar will be visible in a Room. If the field has the value of "1" a link to the Calendar will be displayed in the sidebar h_SetSecurity This field works in conjunction with the h_ShowSecurity field. It is only valid for Readers and Authors, because Managers must always be able to edit security of a Room. If the field is set to "1" a link to the Security page will be displayed in the sidebar for Readers and Authors (if they select Security in this case they will see only their own information) h_MailDb The name of the database that receives email addressed to this Place.

Detailed Description Paragraph Table (8):
TABLE 8 FIELDS USED TO DEFINE FIELDS Field Name Description h_IsUser Defined h_True means this is a custom form h_PublishInFolder UNID of the folder + ".vertline." + h_FolderStorage name of the folder h_Name "Import" and is related to the h_SystemName field which often has a similar value such as "h_Import". h_FieldLabel Instructional information that might be useful for someone editing this field. Similar to the Static h_FieldType. Containing information to help the user, but

only displayed in edit mode." For example: <script> (h_CurrentSkinType = = `h_Edit`)?"": C(self, `Note: Clicking on the title of this page in its folder or in the sidebar will open the page that it points to. To edit the page again later, click its title in the Index.`); </script>" h_ContainerUNID The UNID of the Form which contains this field. QuickPlace uses a Design Note to create forms, each of these having an internal name. The h_ContainerUNID contains the internal name of one of these QuickPlace Forms. h_FieldType There are many different types of Fields. The following types are listed as examples to help understand how Fields work in general. "h_Attachments"= Enables the attaching of files. "h_CalendarControl"= Includes date and time controls and a duration field "h_DateControl"= Date field with date picker widget "h_DateTime"= Contains Date and Time information. "h_DocAuthor"= Contains a Domino Heirachical name of the original Author of the Document. "h_DocCreated"= Creation date of the page. "h_DocModified"= Modified date of the page. "h_DocSize"= Size of the page. "h_NamePopup"= Select listing members of the QuickPlace "h_RichText"= Rich text field. Allowing editing via the rich text editor applet. "h_Serial"= A unique number to identify the document. "h_Static"= Static text, used to provide information about the accompanying field. May also include link to an image. "h_Subject"= The Documents subject. "h_TaskControl"= Used in the Task form to insert the task control tool. "h_TextInput"= Simple text equating to the "<input>" field in HTML. "h_TextPopup"= Text select list, equating to the "<select><option>" in HTML. "h_TimeControl"= Select lists for hours, minutes, AM/PM. "h_CalendarControl"= Field containing control tool used in the calendar field. "h_CreateMSExcel"= Field enabling the upload of Excel documents. "h_CreateMSPowerPoint"= Field enabling the upload of PowerPoint documents. "h_CreateMSWord"= Field enabling the upload of Word documents. "h_Import"= Field enabling the upload of imported documents such as HTML. "h_MultipleImport"= Field enabling the upload of multiple documents, such as a series of HTML documents. "h_NotifyIndicator"= Field indicating if members should be notified of the creation of content or their inclusion in the Contacts1.nsf.

Detailed Description Paragraph Table (10):
TABLE 10 FIELDS USED TO DEFINE QUICKPLACE PAGES Field Name Description h_Form The QuickPlace form used to create this page. This is not the Domino "Form" field which denotes which form Domino links the file to. The Domino "Form" field will contain "h_PageUI" for virtually all objects in a QuickPlace. h_PageType This field is set to null when the document is a visible document. Only when the object is in design mode do the other values appear: "h_Response" the document is a response to a topic document. This value is only valid in response folders. "h_Revision" this means that the document is being revised, and is not available for public access. "h_Mail" means that the document is a mail document, being either sent or received by QuickPlace. h_Originator The creator of this page. This field contains a full hierarchical name, for example: "CN=David Wyss/OU= QuickPlaceName/OU=QP/O= ServerName". All users have the second OU part of the name set to QP. This is done so that when QuickPlace is used on an Overlay server (QuickPlace and Domino together) QuickPlace can avoid conflicts between Domino registered users and QuickPlace users. h_NameIsBanner Denotes if the page's name should be displayed as a banner. If it is to be displayed as a banner, this field contains the value "1". Setting this field is done when the user clicks on the "Show the title, author and date on page?" checkbox.

Detailed Description Paragraph Table (16):

```
Set dirPlace = New NotesDbDirectory( g_sServerName ) Set ndbPlace =
dirPlace.GetFirstDatabase( DATABASE ) sNdbPlaceFilepath = ndbPlace.FilePath If
Instr (1, Lcase ( sNdbPlaceFilepath ),
.vertline.QuickPlace.backslash.millennia.backslash.pagelibrary.vertline. ) Then
```

Detailed Description Paragraph Table (25):
TABLE 24 Folders, What's New, Search Results, Tasks (list view) CSS Selector Description and Notes .h-folderBanner-bg Background of folder banner. Note 1. .h-folderBanner-text Text in folder banner. Note 1. a.h-folderBanner-text Anchors in

folder banner. Note 1. .h-folderBannerSelected-text Text of selected ("current") item in folder banner. Note 1. a.h-folderBannerSelected-text Selected anchor in folder banner. Note 1. .h-folderItem-bg Background of items listed in folder. Note 1. .h-folderItem-text Text of items listed in folder. Note 1. a.h-folderItem-text Anchor listed in folder. Note 1. .h-folderCompact-text Compact text of item listed in folder. Note 1. .h-folderAbstract-text Abstract text of item listed in folder. .h-folderBar-bg Background of bar to left of a thread. .h-folder-dl { Indentation of responses in response folder. Note: by default, the margin-bottom property is set to 0px to remove unwanted whitespace below indented items in response folders. .h-folderInterspace-bg { } Background color of vertical space between responses. .h-folderInterspace-text { Height of vertical space between responses. Note: use font-size to set the height. .h-folderSpace-text { Height of vertical space between threads. Note: use font-size to set the height. Note 1: This style is used for the banner that displays column titles, as well as other banners in What's New, Search Results, etc.

CLAIMS:

5. The method of claim 1, further comprising the steps of: maintaining said collaboration space as a collection of rooms, each room being implemented as a database; associating a view with each said database; associating a folder with each said view; periodically or upon demand, copying said view to said folder; and generating said report by opening and displaying contents of said folder.

8. The method of claim 5, said copying step further being executed whenever said database is updated.

25. The system of claim 24, further comprising: said collaboration space being a collection of rooms, each room implemented as a database; a view associated with each said database; a folder associated with each said view for receiving a copy of said view periodically or upon demand; and said report being generated by opening and displaying contents of said folder.

28. The program storage device of claim 27, said method steps further comprising: maintaining said collaboration space as a collection of rooms, each room being implemented as a database; associating a view with each said database; associating a folder with each said view; periodically or upon demand, copying said view to said folder; and generating said report by opening and displaying contents of said folder.

Previous Doc Next Doc Go to Doc#